SMARTQUIZZER: ADAPTIVE AI POWERED QUIZ GENERATION ENGINE

MENTOR:
P.SATHYA Ma’am

TEAM MEMBERS:
Rajesh Raftar
Nikhil Puppalwar
Tanu Tyagi
S Thejaswini

TABLE OF CONTENTS:

INTRODUCTION

PROBLEM STATEMENT

PROJECT OVERVIEW

TABLE OF CONTENTS:

KEY FEATURES

TECHNOLOGY STACK

SYSTEM ARCHITECTURE

TABLE OF CONTENTS:

APPLICATION FLOW

AI FEEDBACK SYSTEM

RESULTS AND ANALYSIS

TABLE OF CONTENTS:

10 SCREENSHOTS

11 CONCLUSION

12 FUTURE ENHANCEMENTS

INTRODUCTION:

- Traditional quiz tools are **static** and do not adapt to learner skill.
- Students struggle to find quizzes that match their **difficulty level** or **learning pace**.
- **AI SmartQuizzer** solves this by generating **personalized, adaptive quizzes** from any learning material.
- Built using Groq's **LLaMA 3.1** model for intelligent question creation.

PROBLEM STATEMENT:

Learners face challenges like:

- Generic, non-adaptive quiz systems
- No personalization based on performance
- Limited variety of question types
- Difficulty analyzing learning progress

Need: A dynamic, data-driven quiz system that adjusts to user performance.

PROJECT OVERVIEW:

AI SmartQuizzer enables:

- Uploading learning materials (PDF, Text, URL)
- Automatic quiz creation using AI
- Difficulty adjustment based on user performance
- Performance analytics with charts and statistics
- User accounts with saved preferences

KEY FEATURES:

- Content Ingestion (PDF/Text/URL)
- AI-generated Questions (MCQ, T/F, Short Answer, Fill in the Blanks)
- Adaptive Difficulty Engine
- Analytics Dashboard
- Admin Panel for Moderation
- Secure Authentication System

TECHNOLOGY STACK:

Backend: Python Flask
AI: Groq API – LLaMA 3.1
Database: SQLite + SQLAlchemy ORM
Frontend: HTML, CSS, JavaScript
Visualization: Chart.js
PDF Parsing: PyMuPDF

SYSTEM ARCHITECTURE:

Three-tier architecture:

Presentation Layer

- HTML (Jinja2 templates)
- JavaScript (quiz logic, real-time updates)
- Chart.js for analytics

Application Layer

- Flask backend
- AI-based question generation
- Adaptive learning logic

Data Layer

- SQLAlchemy ORM
- SQLite Database

APPLICATION FLOW:

- User interacts via UI
- Flask routes receive request
- Backend triggers business logic
- AI model or database performs processing
- Response (HTML/JSON) sent back to user

AI FEEDBACK SYSTEM:

Feedback includes:

- Encouragement message
- Weak areas & concepts needing improvement
- Study recommendations
- Motivation to continue learning

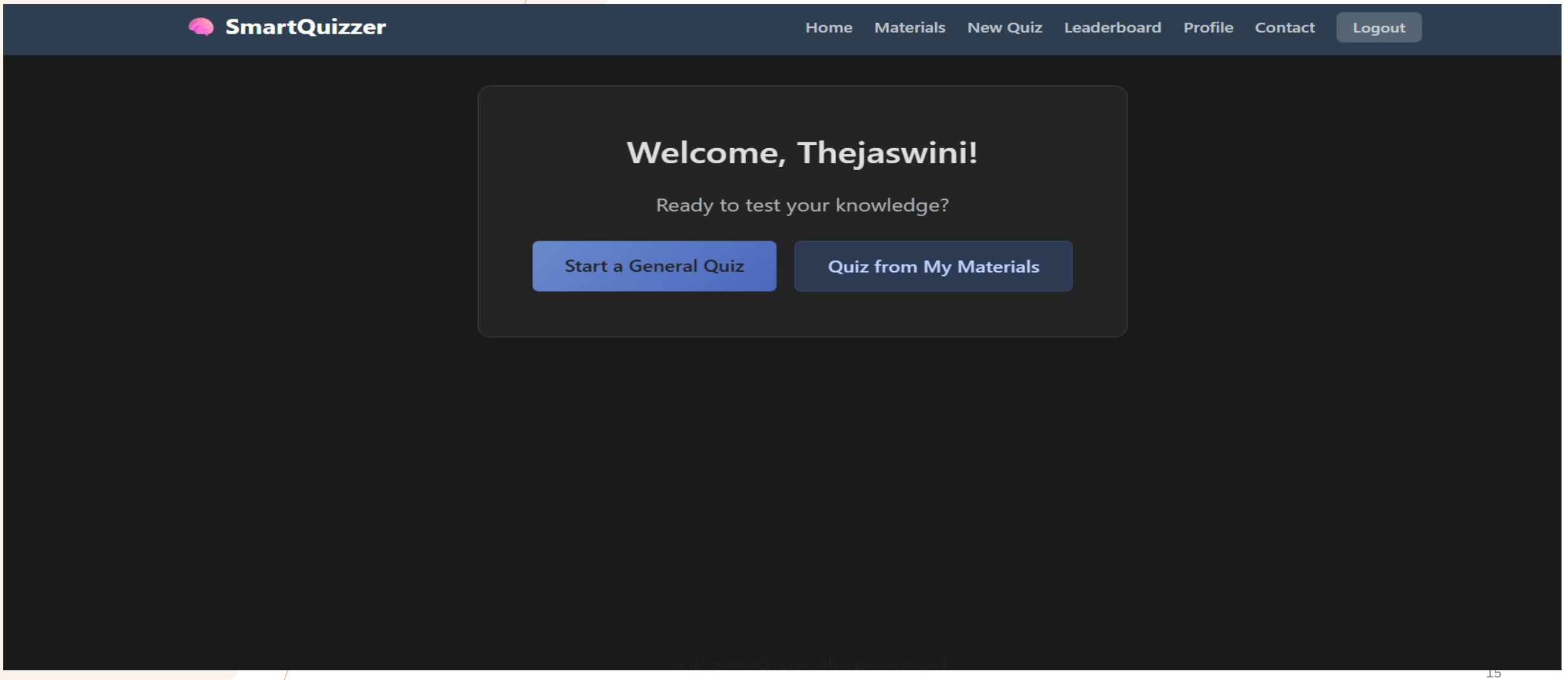
Appears at the end of each quiz attempt.

RESULTS AND ANALYSIS:

Displays:

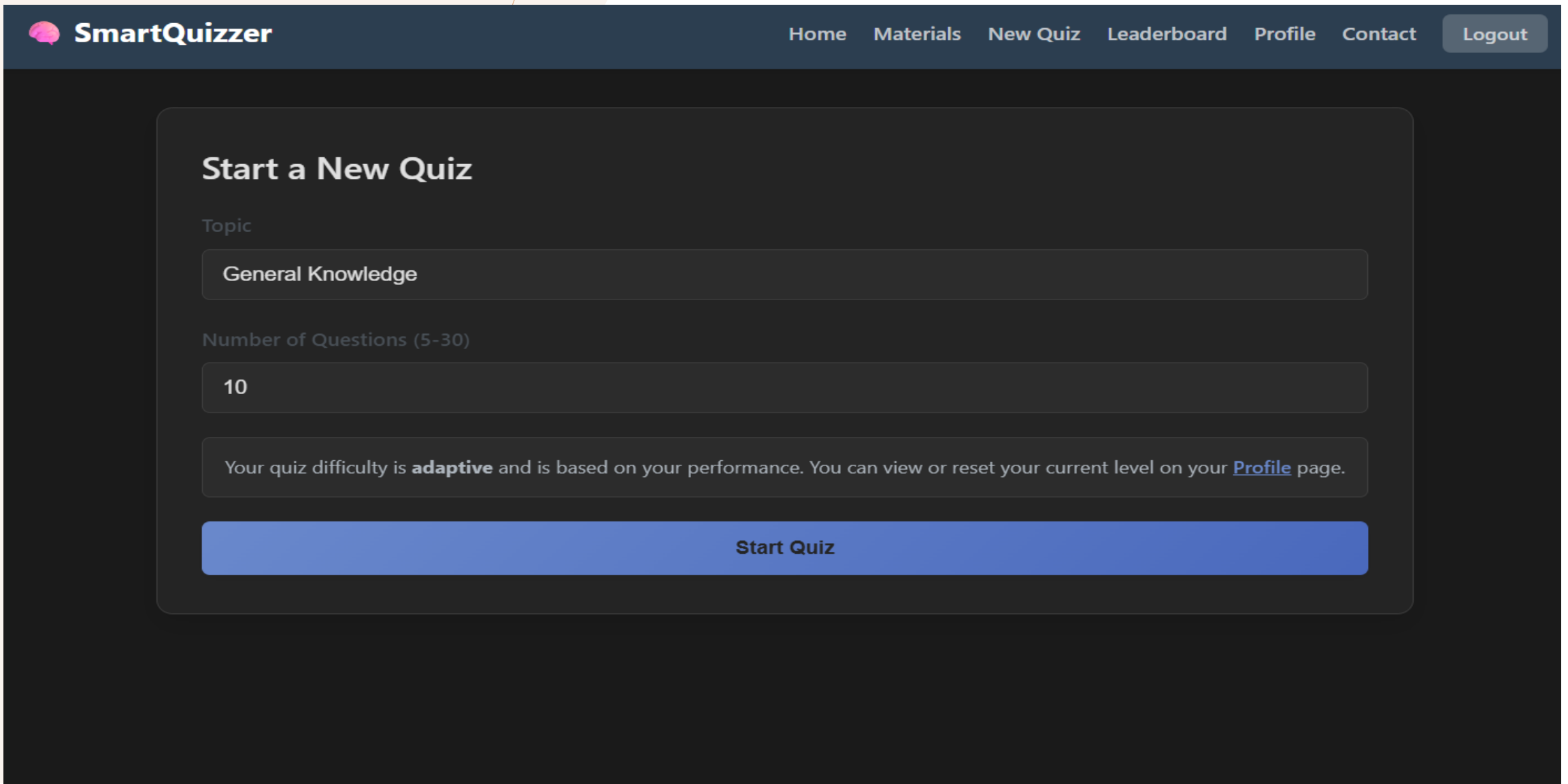
- Total quizzes taken
- Total questions attempted
- Overall accuracy
- Avg. time per question
- Score trends over time (line chart)
- Subject performance (bar chart)
- Score distribution (doughnut chart)
- Personal best attempt summary

SCREENSHOTS:

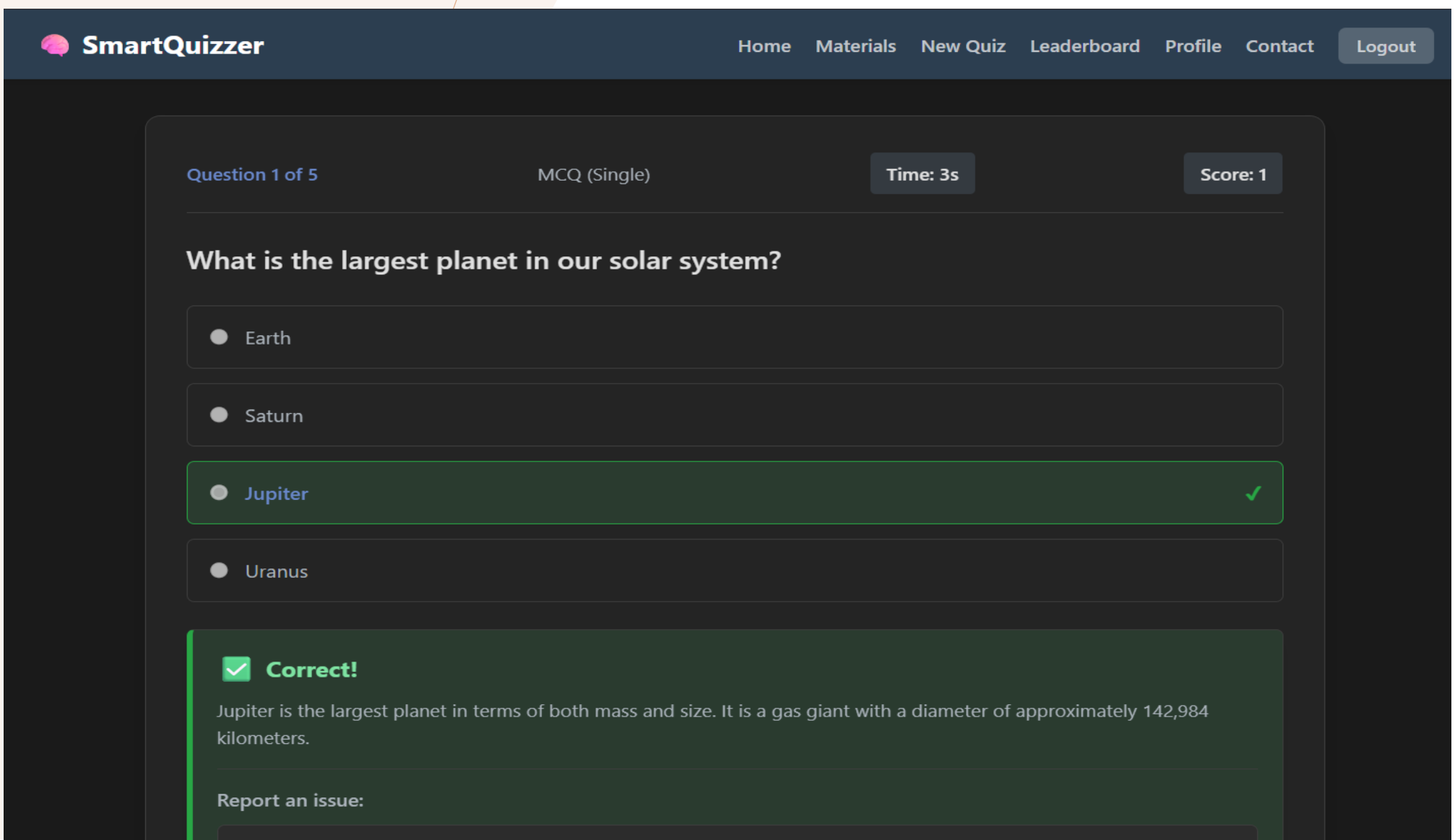


© 2025 SmartQuizzer. All rights reserved

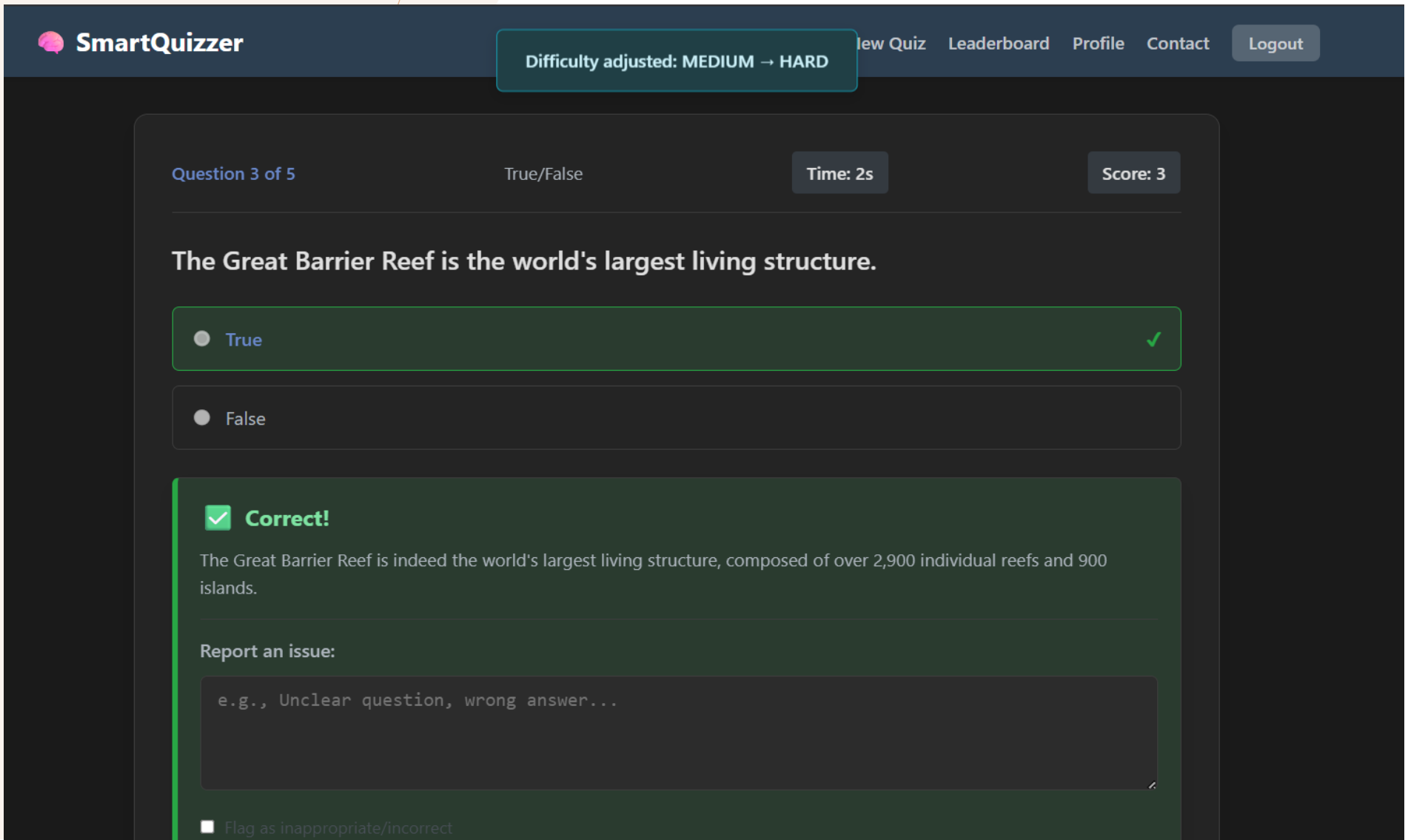
SCREENSHOTS:



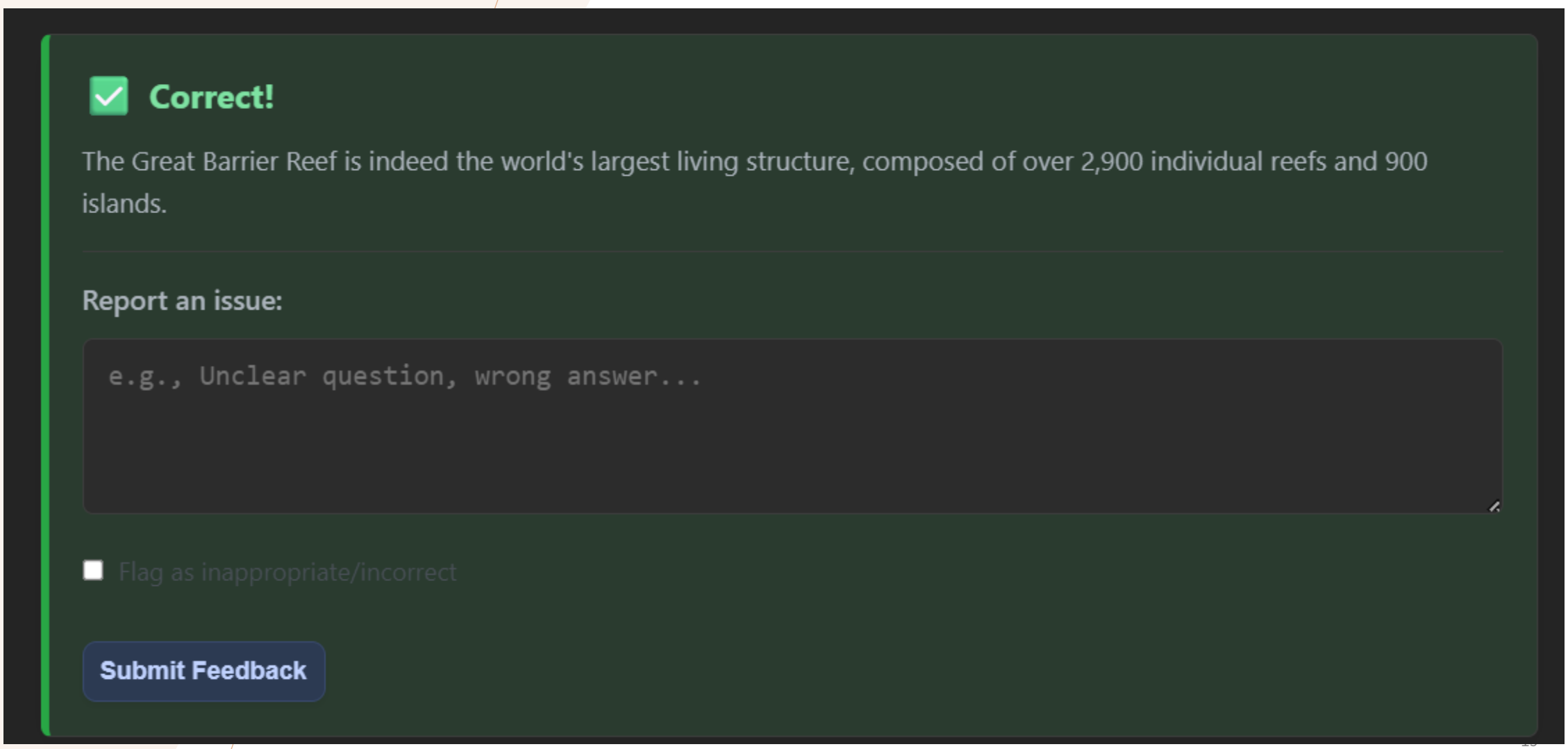
SCREENSHOTS:



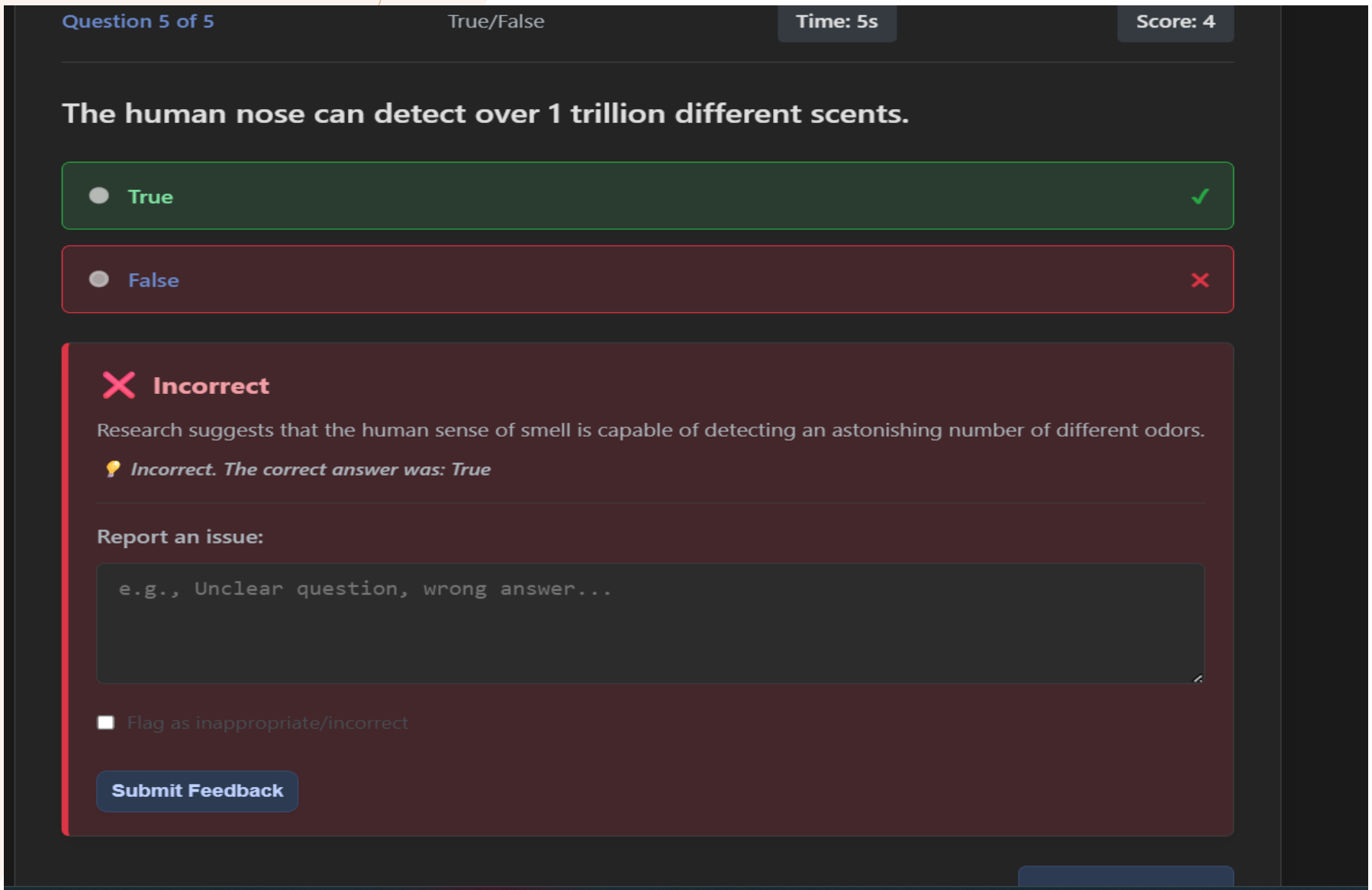
SCREENSHOTS:



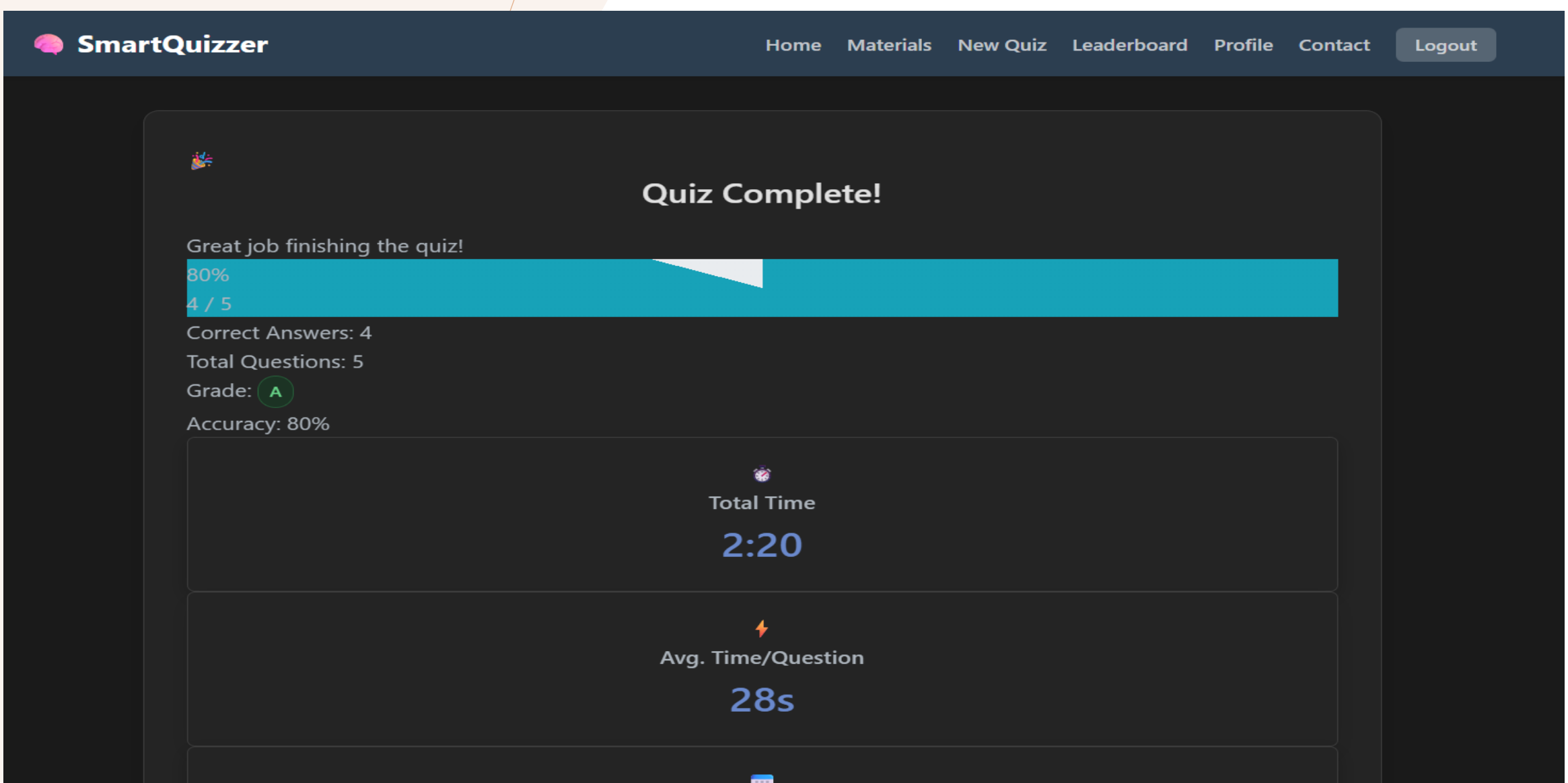
SCREENSHOTS:



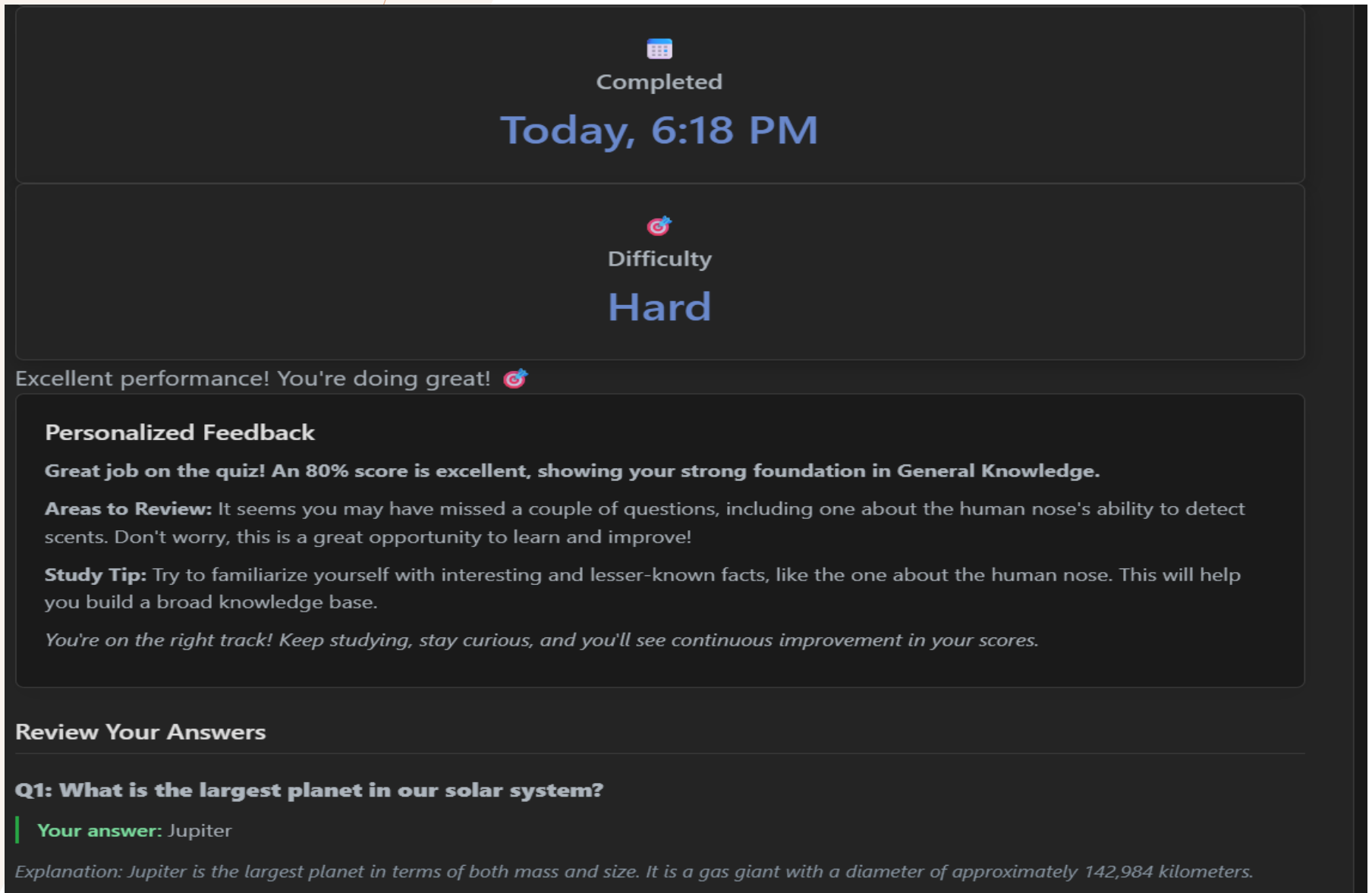
SCREENSHOTS:



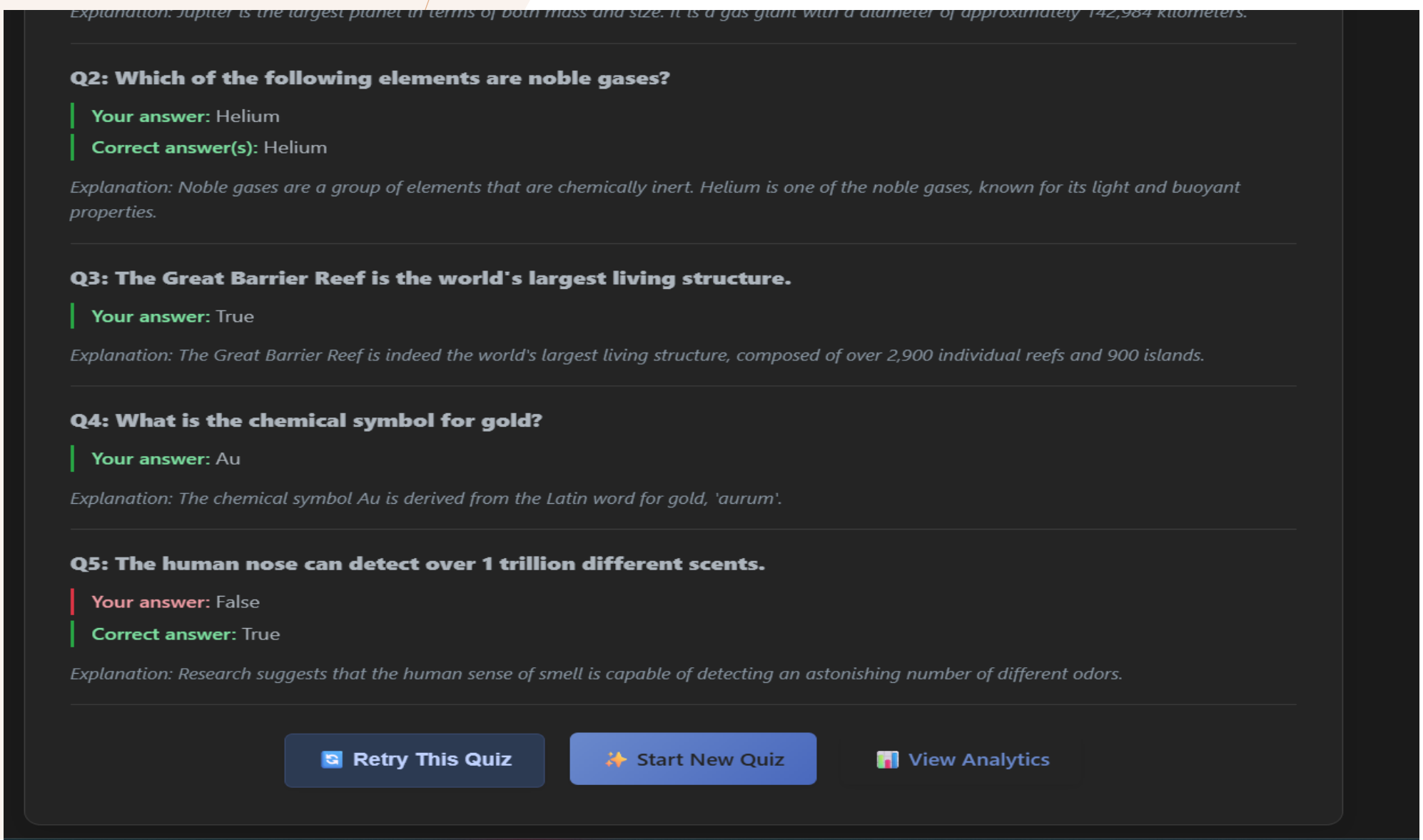
SCREENSHOTS:



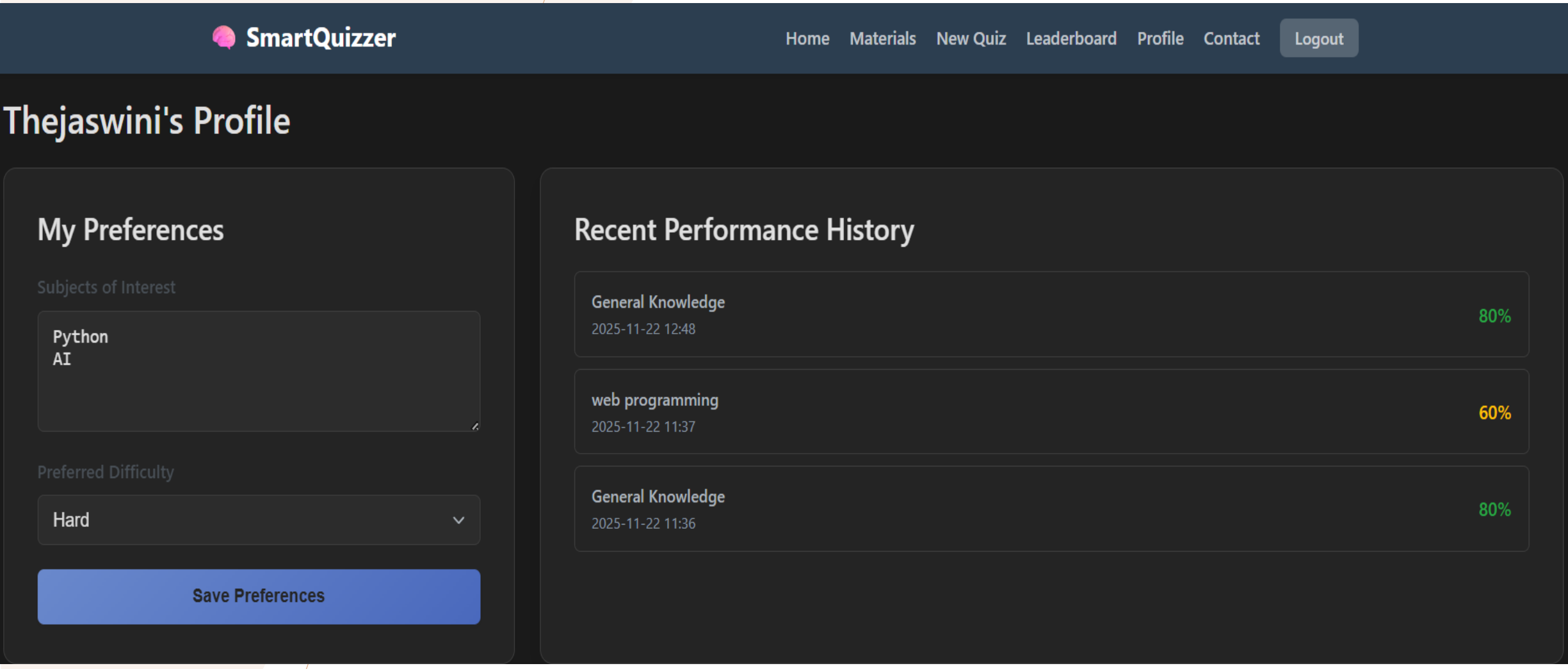
SCREENSHOTS:



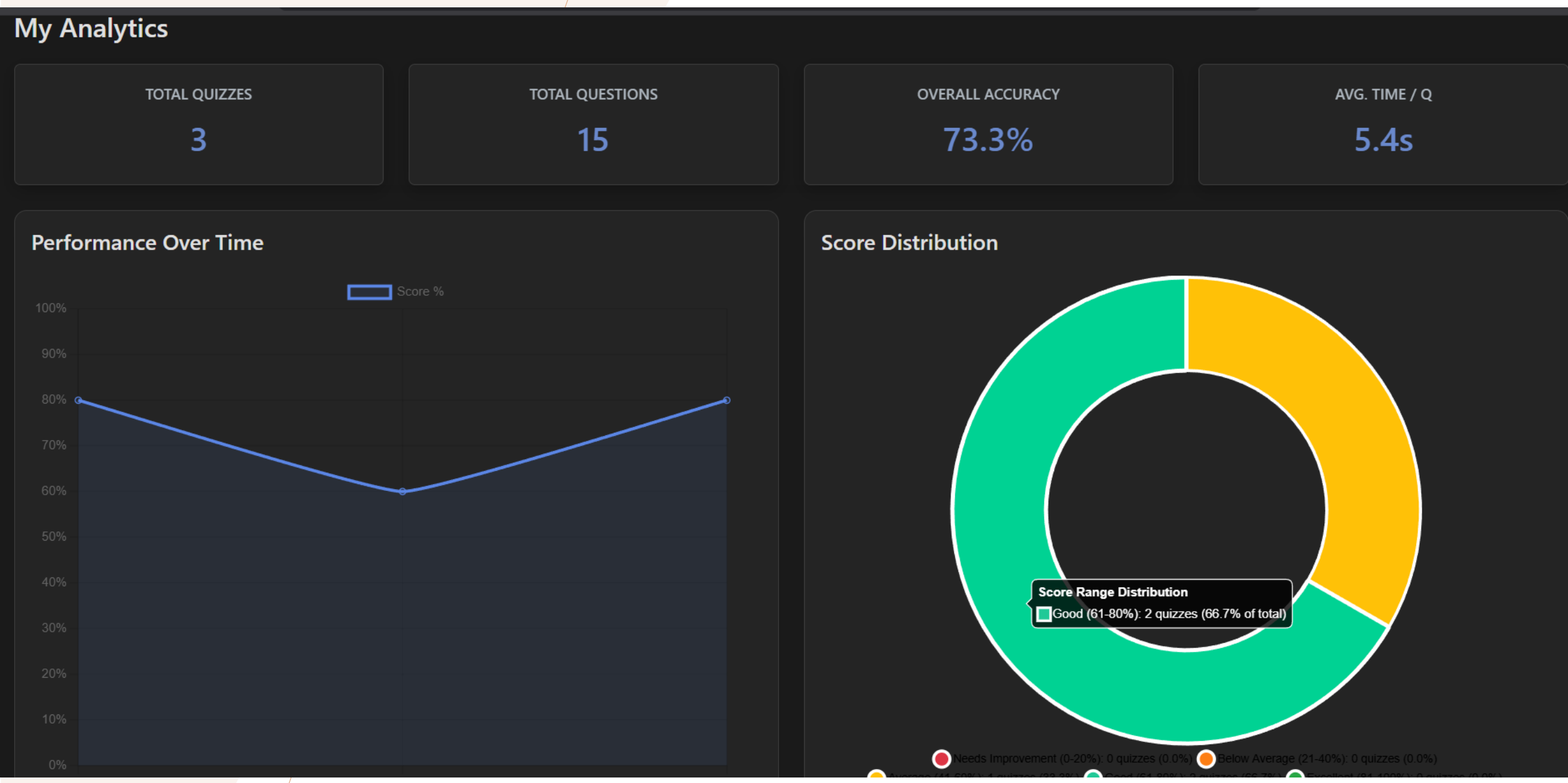
SCREENSHOTS:



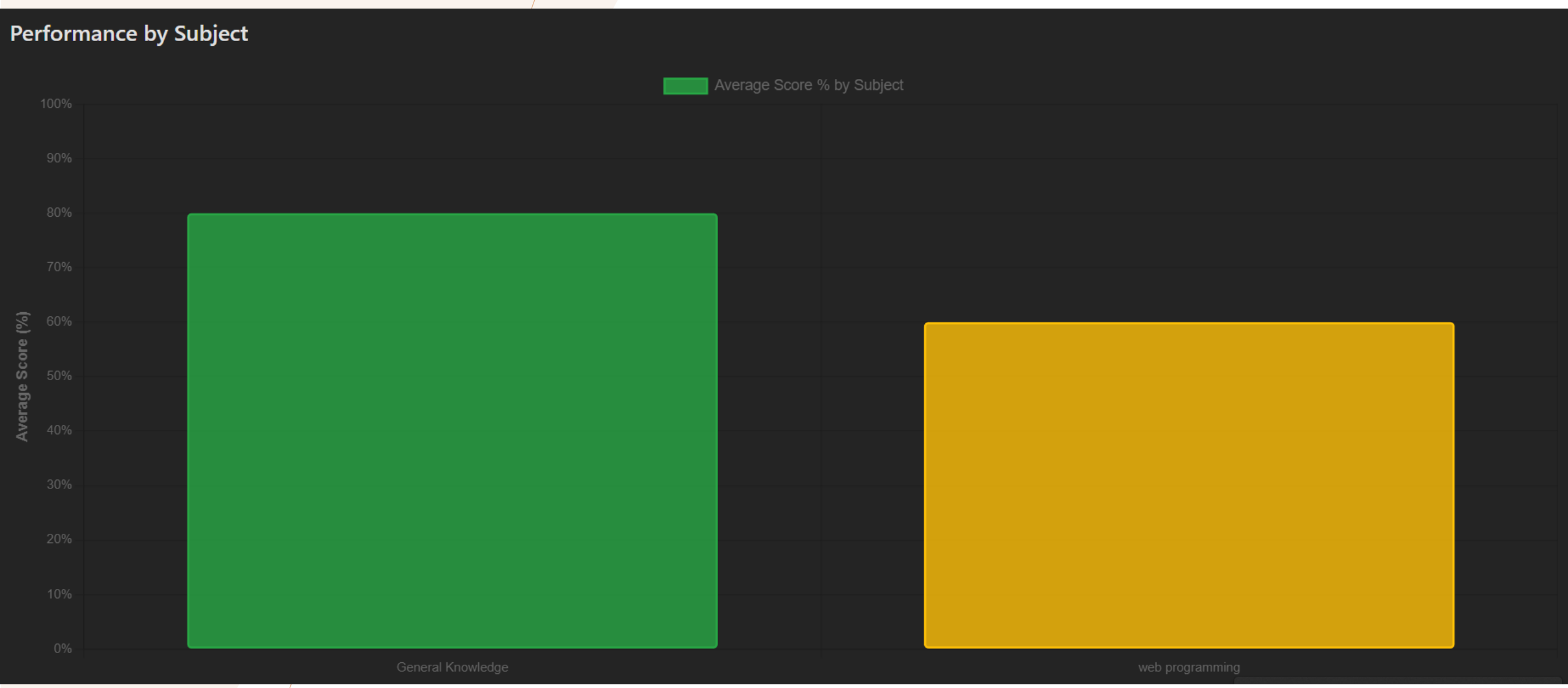
SCREENSHOTS:



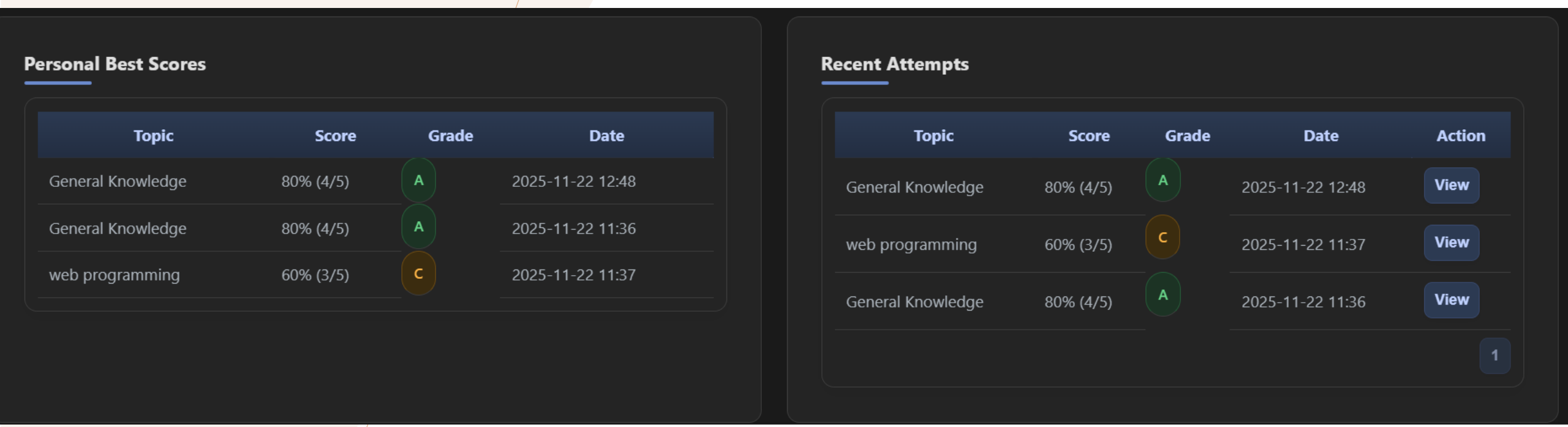
SCREENSHOTS:



SCREENSHOTS:



SCREENSHOTS:



Personal Best Scores

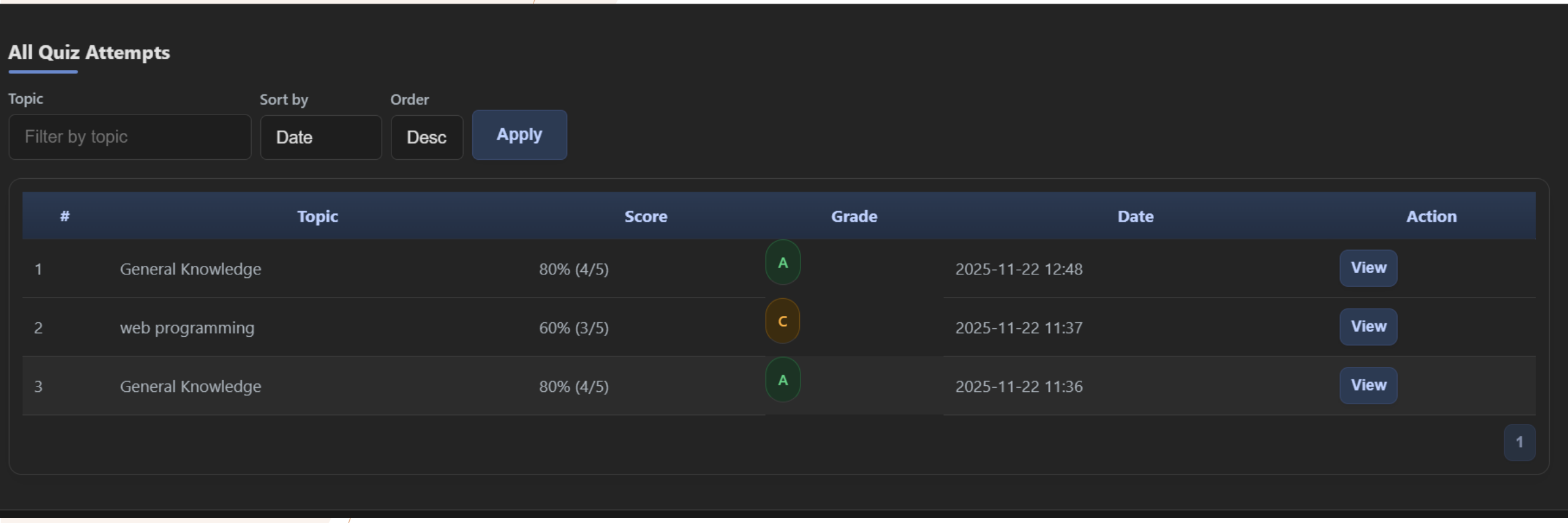
Topic	Score	Grade	Date
General Knowledge	80% (4/5)	A	2025-11-22 12:48
General Knowledge	80% (4/5)	A	2025-11-22 11:36
web programming	60% (3/5)	C	2025-11-22 11:37

Recent Attempts

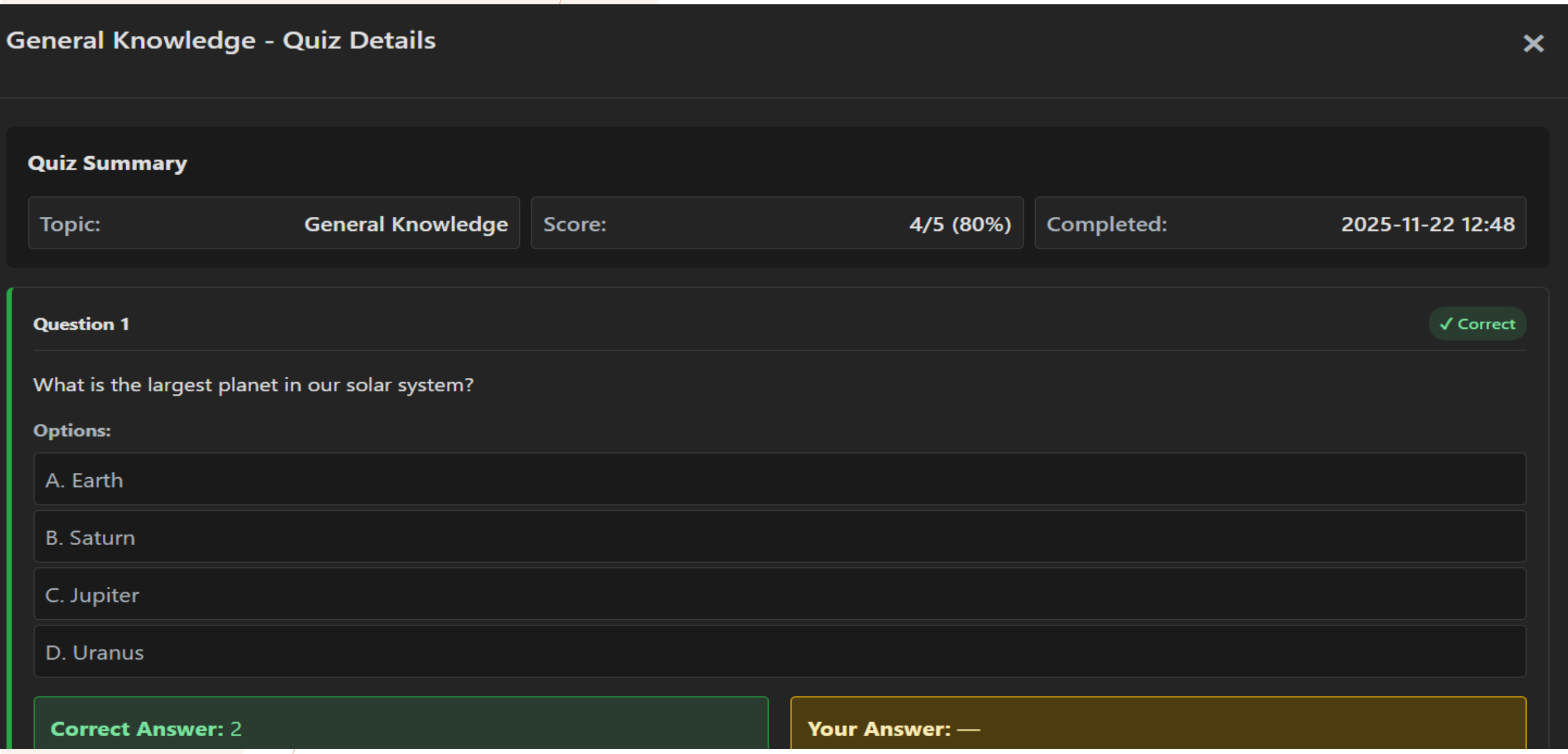
Topic	Score	Grade	Date	Action
General Knowledge	80% (4/5)	A	2025-11-22 12:48	View
web programming	60% (3/5)	C	2025-11-22 11:37	View
General Knowledge	80% (4/5)	A	2025-11-22 11:36	View

1

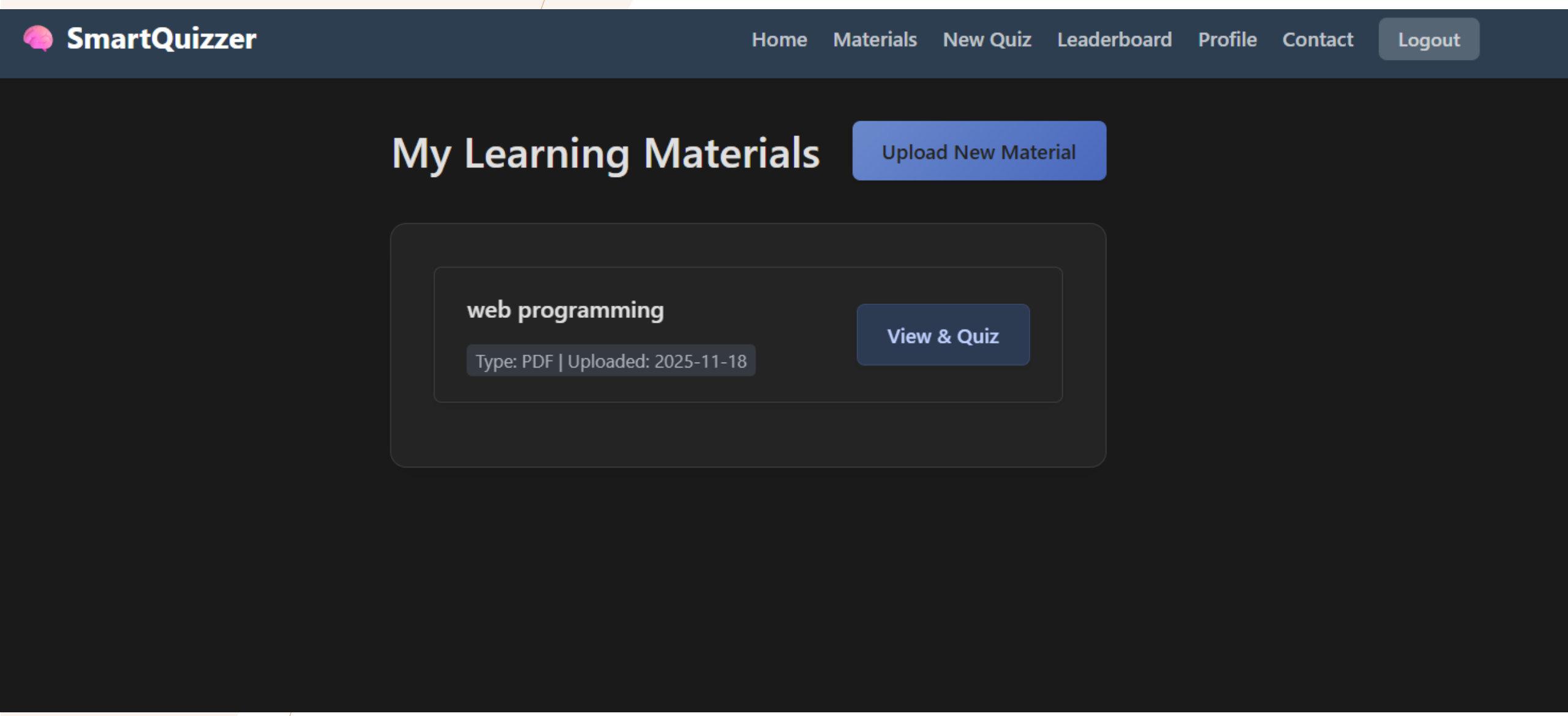
SCREENSHOTS:



SCREENSHOTS:



SCREENSHOTS:



CONCLUSION:

- AI SmartQuizzer provides **personalized, adaptive learning**
- Combines modern AI + web tech for dynamic quiz generation
- Helps students learn efficiently with real-time analytics
- Designed with modular architecture for scalability
- Suitable for both self-learners & institutions

FUTURE ENHANCEMENTS:

- Add web scraping for URL-based content ingestion
- Improve feedback clarity using deeper LLM reasoning
- Add gamification (badges, levels)
- Support for more advanced LMS integration
- Voice-based quiz interaction

THANK YOU